

voter digital ID, assigned by identity services **130**, a voter public key and private key, assigned by identity services **130**, and a token to vote assigned by tokenizer vault **133**.

**[0073]** In some embodiments, the voting software system **200** can register VSO **201** to vote in an election software object **202** (“ESO **202**”). An ESO **202** is a definition of the basic attributes of the process for filling open positions for “Public Office” in the federal, state, county or municipality, or determining the course of action to be taken on public policy issues through a voter referendum, ballot measure, and the like. In some embodiments, the ESO **202** can contain information on a specific election such as, an alpha-numeric election ID, a description of the election, the state and county the election is taking place in, the various candidates, positions, and public referendums on the federal, state, county, local, and municipal positions and issues to be voted on, the date of the election and the status of the election.

**[0074]** In some embodiments, voting software system **200** can associate ESO **202** with a ballot template software object **203** (BTSO **203**). In some embodiments, BTSO **203** elaborates the details of an election. For example BTSO **203** can be a State/County specific template showing the various categories and sub-categories of the open positions and the candidates (including their affiliation) who are running for those positions, and any “ballot measures” seeking citizen referendum. In some embodiments, a unique identifier is assigned to each ballot based on the election and the voter receiving the ballot and is then used to mail that ballot to a voter.

**[0075]** In some embodiments, voting software system **200** uses county registrar software object **204** (CRSO **204**) to define an election and create a ballot template. In some embodiments, CRSO **204** helps to define the elections and create the ballot. CRSO **204** varies by what state the blockchain powered vote by mail system is implemented in. In some embodiments, the CRSO **204** can help the actual physical county registrar certify the official lists of candidates running for state offices, advise candidates and local elections officials on the qualifications and requirements for running for office, provide guidance on how candidates can select acceptable candidate ballot designations, determine the order in which candidates are placed on the ballot, track and certify ballot initiatives, coordinate the tabulation of the votes from each county on election night, and use its voter registration and outreach team to produce voter registration forms, voter information publications, and encourage people to register and vote.

**[0076]** In some embodiments, voting software system **200** can receive input from an actual voter and can then “cast” or create ballot software object **205** (BSO **205**), which is a specific instance of BTSO **203**. BSO **205** is completed ballot template **203** and is associated with the VSO **201** of the voter that provided the input that was used to fill out BSO **205**. In some embodiments, BSO **205** contains a collection of vote software objects **206**, which represent the actual votes cast by the voter that corresponds to a specific VSO **201**.

**[0077]** In some embodiments, the voting software system **200** can use notary software object **207** (NSO **207**) to certify that BSO **205** was correctly cast. In some embodiments, the NSO **207** certifies that BSO **205** was correctly cast by verifying a hash provided with the BSO **205** with its own computation.

**[0078]** In some embodiments, the NSO **207** will also certify results software object **208** (RSO **208**), which is an

aggregate of all of the casted votes and represents the result of the election. In some embodiments, the NSO **207** similarly certifies RSO **208** by verifying a hash provided with the RSO **208** with its own computation. RSO **208** is calculated by the voting software system **200** using the accumulator software object **209** (ASO **209**). ASO **209** appropriately buckets each vote received to the receiving candidate. ASO ensures each vote that is recorded is counted properly and can summarize the votes received by various categories.

**[0079]** FIG. 3 shows a software hierarchy diagram for the various ways different users can interact with the blockchain access layer **101** through user interface **131**. In some embodiments, different users have access to different functions that they can use to perform actions through blockchain access layer **101**. At the highest level, all types of users can interact with blockchain access layer **101** through the functions contained through an interface **301** software object. In some embodiments, interface **301** is a software object contained within user interface **131** and/or utilize user interface **131**. Interface **301** allows users to access basic blockchain functions. For example, interface **301** allows all users to verify the connection to the blockchain, instantiate an API that allows control of the blockchain access layer **101** and also allows the users to interact with databases **150**. In some embodiments, this interface can be a BaseWeb3 type interface.

**[0080]** The next hierarchical level is a participant **302** software object. Participant **302** is a software object that interacts with interface **301** to allow users to perform functions common to all users. In some embodiments, participants can use the interface **301** to create an account on the blockchain access layer **101**, create a user on the blockchain access layer **101**, generate a public and private key pair for the user that is used for signing transactions entered onto the blockchain, login to the system, and sign specific transactions. Participants can also come in three categories: voters, registrars, and notaries. Each category can perform additional specific functions for that particular category of participants.

**[0081]** For example, some participants are voters **303**. In some embodiments, voters can register to vote through the system (e.g. request receive an ballot), register for a digital voter ID, cast a digital ballot, scan a token, review their own voting status, and view the electronic ballot that they previously cast. In some embodiments, the voters **303** can view the electronic ballots they previously cast by using the token associated with their ballots by accessing their ballot in voter-ballot database **154** through blockchain access layer **101** and user interface **131**. In some embodiments, the voters **303** can also use the user interface to track the progress of physical ballots as it traverses the mail system from both the election office and its return to the election office.

**[0082]** In some embodiments, participants **302** can also be election registrars **304**. In some embodiments, election registrars **304** can register their election on the blockchain and define the election, including what date and time, what positions are open, who is running for those elections, and what ballot measures are on the ballot. Election registrars **304** can also create a template ballot for the election and view the voter status for each voter (e.g. who has voted or what they have voted on). In some embodiments, the registrars can access a list of which voters voted via the blockchain without accessing the actual votes the voters cast.